

SSSSSSSSSSSS	000000000	RRRRRRRRRRR	TTTTTTTTTTTTTTT	33333333333	222222222
SSSSSSSSSSSS	000000000	RRRRRRRRRRR	TTTTTTTTTTTTTTT	33333333333	222222222
SSSSSSSSSSSS	000000000	RRRRRRRRRRR	TTTTTTTTTTTTTTT	33333333333	222222222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSSSSSSSSS	000	000	RRRRRRRRRRR	TTT 333	222 222
SSSSSSSSSS	000	000	RRRRRRRRRRR	TTT 333	222 222
SSSSSSSSSS	000	000	RRRRRRRRRRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSS	000	000	RRR RRR	TTT 333	222 222
SSSSSSSSSS	000000000	RRR RRR	TTT 333	33333333333	22222222222222
SSSSSSSSSS	000000000	RRR RRR	TTT 333	33333333333	22222222222222
SSSSSSSSSS	000000000	RRR RRR	TTT 333	33333333333	22222222222222

\*\*FILE\*\* ID\*\*SORENTRY

J 6

SSSSSSSS 000000 RRRRRRRR EEEEEEEEEE NN NN TTTTTTTT RRRRRRRR YY YY  
SSSSSSSS 000000 RRRRRRRR EEEEEEEEEE NN NN TTTTTTTT RRRRRRRR YY YY  
SS 00 00 RR RR EE NN NN TT RR RR YY YY  
SS 00 00 RR RR EE NN NN TT RR RR YY YY  
SS 00 00 RR RR EE NNNN NN TT RR RR YY YY  
SS 00 00 RR RR EE NNNN NN TT RR RR YY YY  
SSSSSS 00 00 RRRRRRRR EEEEEEEE NN NN TT RRRRRRRR YY YY  
SSSSSS 00 00 RRRRRRRR EEEEEEEE NN NN TT RRRRRRRR YY YY  
SS 00 00 RR RR EE NN NNNN TT RR RR YY YY  
SS 00 00 RR RR EE NN NNNN TT RR RR YY YY  
SS 00 00 RR RR EE NN NN TT RR RR YY YY  
SS 00 00 RR RR EE NN NN TT RR RR YY YY  
SSSSSSSS 000000 RR RR EEEEEEEEEE NN NN TT RR RR YY YY  
SSSSSSSS 000000 RR RR EEEEEEEEEE NN NN TT RR RR YY YY

SOR  
V04

1 0001 0 MODULE SORSENTRY(MAIN=SORENTRY  
2 0002 0 IDENT = 'V04-000'  
3 0003 0 ) =  
4 0004 1 BEGIN  
5 0005 1  
6 0006 1 \*\*\*\*\*  
7 0007 1 \*  
8 0008 1 \* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
9 0009 1 \* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
10 0010 1 \* ALL RIGHTS RESERVED.  
11 0011 1 \*  
12 0012 1 \* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
13 0013 1 \* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
14 0014 1 \* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
15 0015 1 \* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
16 0016 1 \* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
17 0017 1 \* TRANSFERRED.  
18 0018 1 \*  
19 0019 1 \* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
20 0020 1 \* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
21 0021 1 \* CORPORATION.  
22 0022 1 \*  
23 0023 1 \* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
24 0024 1 \* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
25 0025 1 \*  
26 0026 1 \*  
27 0027 1 \*\*\*\*\*  
28 0028 1  
29 0029 1  
30 0030 1 ++  
31 0031 1  
32 0032 1 FACILITY: VAX SORT/MERGE  
33 0033 1  
34 0034 1 ABSTRACT:  
35 0035 1  
36 0036 1 This module contains the main entry to the sort/merge utility.  
37 0037 1  
38 0038 1 ENVIRONMENT: VAX/VMS user mode  
39 0039 1  
40 0040 1 AUTHOR: Peter D Gilbert, CREATION DATE: 07-Jan-1982  
41 0041 1  
42 0042 1 MODIFIED BY:  
43 0043 1  
44 0044 1 T03-015 Original  
45 0045 1 T03-018 Change stat names to SOR\$K\_xxx. PDG 4-Jan-1983  
46 0046 1 T03-019 Return with the worst severity we've seen. PDG 14-Jan-1983  
47 0047 1 T03-020 Change "work file size used" to "work file allocation".  
48 0048 1 PDG 27-Jan-1983  
49 0049 1 T03-021 Changed the name of SOR\$STAT. PDG 3-Mar-1983  
50 0050 1 T03-022 Reformat statistics, removing MBC and MBF. PDG 8-Jul-1983  
51 0051 1 T03-023 Remove "-11" from statistics. PDG 10-Nov-1983  
52 0052 1 --

```
54 0053 1 LIBRARY 'SYSSLIBRARY:STARLET';  
55 0054 1 LIBRARY 'SYSSLIBRARY:XPORT';  
56 0055 1  
57 0056 1 %IF %DECLARED(%QUOTE SDESCRIPTOR) %THEN UNDECLARE %QUOTE SDESCRIPTOR; %FI  
58 0057 1  
59 0058 1 LINKAGE JSB_ONE_STAT = JSB (REGISTER=1): NOTUSED(2,3,4,5,6,7,8,9,10,11);  
60 0059 1  
61 0060 1  
62 0061 1 FORWARD ROUTINE  
63 0062 1 COND HAND, ! Handle exception conditions  
64 0063 1 SOR ERROR, ! Issue an error diagnostic  
65 0064 1 SORSENTRY ! Main entry point  
66 0065 1 INIT STATS, ! Get initial statistics  
67 0066 1 ONE STAT: JSB_ONE_STAT, ! Get one statistic  
68 0067 1 PRINT_STATS: ! Print sort/merge statistics  
69 0068 1  
70 0069 1 EXTERNAL ROUTINE  
71 0070 1 SOR$COMMAND, ! Parse command line  
72 0071 1 SOR$OUTPUT, ! Output text  
73 0072 1 SOR$SORT MERGE: ADDRESSING_MODE(GENERAL), ! Sort the stuff  
74 0073 1 SOR$END SORT: ADDRESSING_MODE(GENERAL), ! Terminate sort/merge  
75 0074 1 SOR$STAT: ADDRESSING_MODE(GENERAL), ! Get a statistic  
76 0075 1 LIB$FIXUP_FLT: ADDRESSING_MODE(GENERAL),  
77 0076 1 LIB$FIXUP_DEC: ADDRESSING_MODE(GENERAL),  
78 0077 1 LIB$SIGNAL: ADDRESSING_MODE(GENERAL);  
79 0078 1  
80 0079 1 MACRO  
81 0080 1 BASE_ = 0, 0, 0, 0 %:  
82 0081 1  
83 0082 1 EXTERNAL LITERAL  
84 0083 1 SORT$_FACILITY:  
85 0084 1 BIND  
86 0085 1 SORS_SHR_SYSERROR = SHRS_SYSERROR + STSSK_SEVERE + SORT$_FACILITY ^ 16;  
87 0086 1  
88 0087 1 ! FAO string used to output statistics via SYSSPUTMSG.  
89 0088 1  
90 0089 1 ! The following text interacts closely with the code in PRINT_STATS.  
91 0090 1 ! The text can, however, be changed (translated) independent of the code, if  
92 0091 1 ! the control string still uses the same FAO parameters, and text expands to  
93 0092 1 ! no more than 1024 characters (a restriction of the way that the text is  
94 0093 1 ! output), and lines are separated by carriage-return/line-feed pairs.  
95 0094 1  
96 0095 1 ! Note that the use of tab character in the text is avoided, since  
97 0096 1 ! some terminals may not have tab stops at multiples of eight.  
98 0097 1  
99 0098 1 MACRO  
100 L 0099 1 STR_STATS = %EXPAND %STRING(  
101 L 0100 1 %IF %SWITCHES(DEBUG)  
102 L 0101 1 %THEN '!/!18* VAX Sort/Merge !AC Statistics'  
103 L 0102 1 %ELSE '!/!18* VAX Sort/Merge !+Statistics' %FI,  
104 L 0103 1  
105 L 0104 1 '!Records read:!12UL', '!10* Input record length:!9UL',  
106 L 0105 1 '!Records sorted:!10UL', '!10* Internal length:!13UL',  
107 L 0106 1 '!Records output:!10UL', '!10* Output record length:!8UL',  
108 L 0107 1 '!Working set extent:!6UL', '!10* Sort tree size:!14UL',  
109 L 0108 1 '!Virtual memory:!10UL', '!10* Number of initial runs:!6UL',  
110 L 0109 1 '!/Direct I/O:!14UL', '!10* Maximum merge order:!9UL',
```

SORENTRY  
V04-000

: 111 L 0110 1  
: 112 L 0111 1  
: 113 L 0112 1  
: 114 L 0113 1

'!/Buffered I/O:!12UL',  
'!/Page faults:!13UL',  
'!/Elapsed time: !14%t',  
) %;

M 6  
16-Sep-1984 00:23:12 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 13:10:43 [SORT32.SRC]SORENTRY.B32;1

Page 3  
(2)

'!10\* Number of merge passes:!6UL',  
'!10\* Work file allocation:!8UL',  
'!7\* Elapsed CPU:!6\* !14%T',

SOF  
VO4

: F

```
116 0114 1 ; Besides information that is stored in the context area for statistics,  
117 0115 1 ; a save area is used to store initial values of some statistics.  
118 0116 1  
119 0117 1 FIELD  
120 0118 1     STAT_FIELDS =  
121 0119 1         SET  
122 0120 1             STAT_BUFI0 = [$INTEGER],      ; Buffered I/O count  
123 0121 1             STAT_CPUTIM = [$INTEGER],    ; CPU time  
124 0122 1             STAT_START = [$SUB_BLOCK(2)],   ; Start time (quadword)  
125 0123 1             STAT_DIRIO = [$INTEGER],       ; Direct I/O count  
126 0124 1             STAT_PAGEFLTS = [$INTEGER],    ; Page faults  
127 0125 1             STAT_FREPOVA = [$INTEGER]      ; Free page in P0 space  
128 0126 1         TES;  
129 0127 1 LITERAL  
130 0128 1     STAT_K_SIZE = $FIELD_SET_SIZE; ! Size of save area for statistics  
131 0129 1 MACRO  
132 0130 1     STAT_BLOCK = BLOCK[STAT_K_SIZE] FIELD(STAT_FIELDS) %;  
133 0131 1  
134 0132 1 OWN  
135 0133 1     CONTEXT: LONG,                                ; Context parameter  
136 0134 1     STATS:  STAT_BLOCK,                          ; Block to save statistics  
137 0135 1     BUFI0,  
138 0136 1     CPUTIM: VECTOR[2],  
139 0137 1     DIRIO,  
140 0138 1     PAGEFLTS,  
141 0139 1     FREPOVA,  
142 0140 1     WSEXTENT,  
143 0141 1     SOR_SEV,  
144 0142 1     SOR_STS;  
145 0143 1 BIND  
146 0144 1     ITMLST = UPLIT(  
147 0145 1         WORD(4,JPIS_BUFI0), BUFI0, 0,  
148 0146 1         WORD(4,JPIS_CPUTIM), CPUTIM, 0,  
149 0147 1         WORD(4,JPIS_DIRIO), DIRIO, 0,  
150 0148 1         WORD(4,JPIS_PAGEFLTS), PAGEFLTS, 0,  
151 0149 1         WORD(4,JPIS_FREPOVA), FREPOVA, 0,  
152 0150 1         WORD(4,JPIS_WSEXTENT), WSEXTENT, 0,  
153 0151 1     0);
```

```
155    0152 1 ROUTINE COND_HAND
156    0153 1 (
157    0154 1     SIGVEC: REF BLOCK[,BYTE],
158    0155 1     MCHVEC: REF BLOCK[,BYTE]
159    0156 1     ) =
160    0157 1 !++
161    0158 1
162    0159 1 FUNCTIONAL DESCRIPTION:
163    0160 1
164    0161 1 Condition handler for errors occurring during sort/merge.
165    0162 1 The returned R0 is set to the error message.
166    0163 1
167    0164 1 FORMAL PARAMETERS:
168    0165 1
169    0166 1     SIGVEC.ra.r      The signal vector
170    0167 1     MCHVEC.ra.r    The mechanism vector
171    0168 1
172    0169 1 IMPLICIT INPUTS:
173    0170 1
174    0171 1     NONE
175    0172 1
176    0173 1 IMPLICIT OUTPUTS:
177    0174 1
178    0175 1     NONE
179    0176 1
180    0177 1 ROUTINE VALUE:
181    0178 1
182    0179 1     Status code.
183    0180 1
184    0181 1 SIDE EFFECTS:
185    0182 1
186    0183 1     NONE
187    0184 1
188    0185 1 !--
189    0186 2 BEGIN
190    0187 2
191    0188 2 ! If we are unwinding, just return
192    0189 2
193    0190 2 IF .SIGVEC[CHFSL_SIG_NAME] EQL SSS_UNWIND THEN RETURN SSS_RESIGNAL;
194    0191 2
195    0192 2
196    0193 2 ! If SSS_ROPRAND, then try using LIB$FIXUP_FLT/DEC
197    0194 2
198    0195 2 IF .SIGVEC[CHFSL_SIG_NAME] EQL SSS_ROPRAND
199    0196 2 THEN
200    0197 2 BEGIN
201    0198 2     BUILTIN
202    0199 2     AP,
203    0200 2     CALLG;
204    0201 2 LOCAL
205    0202 2     SIG_PC:   REF VECTOR[,BYTE],      ! PC of bad instruction
206    0203 2     STATUS:
207    0204 2
208    0205 2     SIG_PC = .VECTOR[SIGVEC[BASE_], .SIGVEC[CHFSL_SIG_ARGS]-1];
209    0206 2
210    0207 2     ! Repair the operand, based on the opcode
211    0208 2
```

74  
73  
55  
63  
20  
30  
2F

```

212      0209 4   STATUS = (SELECTONE .SIG_PC[0] OF
213      0210 4     SET
214      0211 4     [OPS_CVTTP, OPS_CVTSP];
215      0212 4     [OPS_CMPF, OPS_MP, OPS_ESCD];
216      0213 4     [OTHERWISE];
217      0214 3     TES);
218      0215 3
219      0216 3   IF .STATUS EQL SSS_NORMAL
220      0217 3   THEN
221      0218 4     BEGIN
222      0219 4       ! We managed to repair the problem.
223      0220 4       ! However, we should let the user know that an error occurred.
224      0221 4
225      0222 4     EXTERNAL LITERAL SORS_ROPRND;
226      0223 4     LIB$SIGNAL(SORS_ROPRND);
227      0224 4     RETURN SSS_NORMAL;
228      0225 4     END;
229      0226 4
230      0227 2     END;

231      0228 2
232      0229 2
233      0230 2   ! Set the returned R0 value
234      0231 2
235      0232 2   MCHVEC[CHFSL_MCH_SAVR0] = .SIGVEC[CHFSL_SIG_NAME];
236      0233 2
237      0234 2   ! Hang onto the worst error we've seen
238      0235 2
239      0236 2   BEGIN
240      0237 2     BIND CVT_SEV = UPLIT BYTE(2,0,3,1,4,5,6,7): VECTOR[,BYTE];
241      0238 2     LOCAL SEV;
242      0239 2     SEV = :CVT_SEV[BLOCK[SIGVEC[CHFSL_SIG_NAME], STSSV_SEVERITY;, ,BYTE]];
243      0240 2     IF .SEV GTRU .SOR_SEV
244      0241 3     THEN
245      0242 4       BEGIN
246      0243 4         SOR_SEV = .SEV;
247      0244 4         SOR_STS = .SIGVEC[CHFSL_SIG_NAME] OR STSSM_INHIB_MSG;
248      0245 3       END;
249      0246 2
250      0247 2
251      0248 2
252      0249 2   ! Resignal the error. If the severity of the error is Success, Info,
253      0250 2   ! Warning, or Error, execution will continue.
254      0251 2
255      0252 2   RETURN SSS_RESIGNAL;
256      0253 2
257      0254 1   END;

```

```

.TITLE SORSENTRY
.IDENT \V04-000\
.PSECT SPLITS,NOWRT,NOEXE,2

040C 0004 00000 P.AAA: .WORD 4, 1036
00000000, 00004 .ADDRESS BUFI0
00000000 00008 .LONG 0
0407 0004 0000C .WORD 4, 1031

```

SORSENTRY  
V04-000  
3A  
6C  
72  
20  
3A  
30  
6F  
65  
30  
55  
4F  
67  
2F  
72  
69  
31  
20

00000000' 00010 .ADDRESS CPUTIM  
 00000000 00014 .LONG 0  
 040B 0004 00018 .WORD 4, 1035  
 00000000, 0001C .ADDRESS DIRIO  
 00000000 00020 .LONG 0  
 040A 0004 00024 .WORD 4, 1034  
 00000000, 00028 .ADDRESS PAGEFLTS  
 00000000 0002C .LONG 0  
 0404 0004 00030 .WORD 4, 1028  
 00000000, 00034 .ADDRESS FREPOVA  
 00000000 00038 .LONG 0  
 0416 0004 0003C .WORD 4, 1046  
 00000000, 00040 .ADDRESS WSEXTENT  
 07 06 05 00000000 00000000 00044 .LONG 0, 0  
 04 01 03 00 02 P.AAB: .BYTE 2, 0, 3, 1, 4, 5, 6, 7  
 .PSECT \$OWNS,NOEXE,2  
  
 00000 CONTEXT:.BLKB 4  
 00004 STATS: .BLKB 32  
 00024 BUFI0: .BLKB 4  
 00028 CPUTIM: .BLKB 8  
 00030 DIRIO: .BLKB 4  
 00034 PAGEFLTS:  
                        .BLKB 4  
 00038 FREPOVA: .BLKB 4  
 0003C WSEXTENT:  
                        .BLKB 4  
 00040 SOR\_SEV: .BLKB 4  
 00044 SOR\_STS: .BLKB 4  
  
 ITMLST= P.AAA  
 CVT\_SEV= P.AAB  
 .EXTRN SOR\$COMMAND, SOR\$OUTPUT  
 .EXTRN SOR\$SORT\_MERGE, SOR\$END\_SORT  
 .EXTRN SOR\$STAT, LIB\$FIXUP\_FLT  
 .EXTRN LIB\$FIXUP\_DEC, LIB\$SIGNAL  
 .EXTRN SORTSFacility, SOR\$\_ROPRAND  
  
 .PSECT \$CODE\$,\$OWRT,2  
  
 0000 00000 COND\_HAND:  
  
 00000920 51 04 AC D0 00002 .WORD Save nothing : 0152  
 8F 04 A1 D1 00006 MOVL SIGVEC, R1 : 0190  
 00000454 8F 04 A1 D1 00010 CMPL 4(R1), #2336  
         4E 12 00018 BEQL 7\$ : 0195  
 50 61 D0 0001A CMPL 4(R1), #1108  
 51 FC A140 D0 0001D MOVL (R1), R0  
 09 61 91 00022 CMPB -4(R1)[R0], SIG\_PC : 0205  
     05 13 00025 BEQL (SIG\_PC), #9 : 0211  
 26 61 91 00027 CMPB (SIG\_PC), #38  
 00000006 00 09 12 0002A BNEQ 2\$ :  
         6C FA 0002C 1\$: CALLG (AP), LIB\$FIXUP\_DEC  
         1D 11 00033 BRB 5\$ : 0212  
 51 8F 61 91 00035 2\$: CMPB (SIG\_PC), #81

		71	8F	OC	13	00039	BEQL	3\$		
				61	91	0003B	CMPB	(SIG_PC), #113		
			FD	8F	06	13	0003F	BEQL	3\$	
				61	91	00041	CMPB	'SIG_PC), #253		
				09	12	00045	BNEQ	4\$		
				6C	FA	00047	3S:	CALLG	(AP), LIB\$FIXUP_FLT	
				02	11	0004E	BRB	5\$		
				50	D4	00050	4S:	CLRL	STATUS	
				50	D1	00052	5S:	CMPL	STATUS, #1	
				11	12	00055	BNEQ	6\$		
				8F	DD	00057	PUSHL	#SORS ROPRAND		
				01	FB	0005D	CALLS	#1, LIB\$SIGNAL		
				50	01	00064	MOVL	#1, R0		
					04	00067	RET			
				50	08	AC	DO	00068	6S:	
				51	04	AC	DO	0006C	MOVL	
				A0	04	A1	DO	00070	SIGVEC, R1	
				03	00	EF	00075	MOVL	4(R1), 12(R0)	
				50	0000'CF	40	9A	0007B	EXTZV	#0, #3, 4(R1), R0
				50	0000'CF	50	D1	00081	MOVZBL	CVT_SEV[R0], SEV
					10	1B	00086	CMPL	SEV, SOR_SEV	
					7S:	BLEQU	8\$			
				0000'CF	0000'CF	50	DO	00088	MOVL	SEV, SOR_SEV
				04	A1	10000000	8F	C9	BISL3	#268435456, 4(R1), SOR_STS
					0918	8F	3C	00098	MOVZWL	#2328, R0
						04	0009D	RET		

: Routine Size: 158 bytes. Routine Base: \$CODE\$ + 0000

```

: 259      0255 1 ROUTINE SOR_ERROR(ERR) =
: 260      0256 1
: 261      0257 1    ++
: 262      0258 1
: 263      0259 1    FUNCTIONAL DESCRIPTION:
: 264      0260 1
: 265      0261 1    This routine signals an error diagnostic.
: 266      0262 1
: 267      0263 1    FORMAL PARAMETERS:
: 268      0264 1
: 269      0265 1    Parameters passed to LIB$SIGNAL.
: 270      0266 1
: 271      0267 1    IMPLICIT INPUTS:
: 272      0268 1
: 273      0269 1    NONE
: 274      0270 1
: 275      0271 1    IMPLICIT OUTPUTS:
: 276      0272 1
: 277      0273 1    NONE
: 278      0274 1
: 279      0275 1    ROUTINE VALUE:
: 280      0276 1
: 281      0277 1    System status (first parameter of signalled status), with the
: 282      0278 1    INHIB_MSG bit set.
: 283      0279 1
: 284      0280 1    SIDE EFFECTS:
: 285      0281 1
: 286      0282 1    The image may be exitted due to the error.
: 287      0283 1
: 288      0284 1    --
: 289      0285 2    BEGIN
: 290      0286 2    BUILTIN
: 291      0287 2    AP,
: 292      0288 2    CALLG;
: 293      0289 2    LOCAL
: 294      0290 2    STATUS;
: 295      0291 2    CALLG(.AP, LIB$SIGNAL);
: 296      0292 2    RETURN .ERR OR STSSM_INHIB_MSG;
: 297      0293 1    END;

```

0000 00000 SOR_ERROR:										
50	00000000G	00	04	AC	10000000	6C	FA	00002	WORD	Save nothing
						8F	C9	00009	CALLG	(AP), LIB\$SIGNAL
							04	00012	BISL3	#268435456, ERR, R0
									RET	

: Routine Size: 19 bytes, Routine Base: \$CODE\$ + 009E

: 0255
: 0291
: 0292
: 0293

SOF  
VO4  
: C

```
0294 1 GLOBAL ROUTINE SORENTRY =  
0295 1  
0296 1 !++  
0297 1  
0298 1 ! FUNCTIONAL DESCRIPTION:  
0299 1  
0300 1 This is the main entry point to the SORT/MERGE utilities.  
0301 1 This routine does the following:  
0302 1  
0303 1 Parse the command line.  
0304 1 Process the specification file.  
0305 1 Use the callable sort/merge routines to finish processing.  
0306 1 Print statistics, if requested.  
0307 1 Release allocated resources.  
0308 1  
0309 1 ! FORMAL PARAMETERS:  
0310 1  
0311 1 ! ! NONE  
0312 1  
0313 1 ! IMPLICIT INPUTS:  
0314 1  
0315 1 ! ! NONE  
0316 1  
0317 1 ! IMPLICIT OUTPUTS:  
0318 1  
0319 1 ! ! NONE  
0320 1  
0321 1 ! ROUTINE VALUE:  
0322 1  
0323 1 ! System status code.  
0324 1  
0325 1 ! SIDE EFFECTS:  
0326 1  
0327 1 ! ! NONE  
0328 1  
0329 1 ! !--  
0330 2 ! BEGIN  
0331 2 ! LOCAL  
0332 2 ! STATISTICS,  
0333 2 ! SORT_FLAG,  
0334 2 ! STATUS;  
0335 2  
0336 2  
0337 2  
0338 2 ! Initialize the severity and message to success  
0339 2  
0340 2 ! SOR_SEV = 0;  
0341 2 ! SOR_STS = SSS_NORMAL;  
0342 2  
0343 2 ! Establish a condition handler  
0344 2  
0345 2 ! (BUILTIN FP: .FP = COND_HAND);  
0346 2  
0347 2  
0348 2  
0349 2 ! Clear the context longword  
0350 2  
0351 2  
0352 2  
0353 2  
0354 2  
0355 2 ! CONTEXT = 0;
```

```
356      0351    2   ! Initialize the statistics
357      0352
358      0353
359      0354
360      0355
361      0356
362      0357
363      0358
364      0359
365      0360
366      0361
367      0362
368      0363
369      0364
370      0365
371      0366
372      0367
373      0368
374      0369
375      0370
376      0371
377      0372
378      0373
379      0374
380      0375
381      0376
382      0377
383      0378
384      0379
385      0380
386      0381
387      0382
388      0383
389      0384
390      0385
391      0386
392      0387
393      0388
394      0389
395      0390
396      0391
397      0392
398      0393
399      0394
400      0395
401      0396
402      0397
403      0398
404      0399
405      0400
406      0401
407      0402
408      0403
409      0404
410      0405
411      0406
412      0407    1

      ! Initialize the statistics
      STATUS = INIT_STATS();
      IF NOT .STATUS THEN RETURN .STATUS;

      ! Call SOR$COMMAND to process the command line, call SPEC_FILE,
      ! call PASS_FILES, and call INIT_SORT or INIT_MERGE.
      ! The context parameter is not referenced by SOR$COMMAND, it is
      ! just passed to the callable interface routines.
      ! SOR$COMMAND sets or clears SORT_FLAG depending on whether
      ! we were invoked for a sort or a merge, respectively.
      ! SOR$COMMAND sets or clears STATISTICS depending on whether
      ! statistics were requested for the sort/merge.

      STATUS = SOR$COMMAND(
          CONTEXT,
          SORT FLAG,
          STATISTICS,
          (BUILTIN AP: .AP));
      IF NOT .STATUS THEN RETURN .STATUS;

      IF .SORT_FLAG
      THEN
          BEGIN
              ! CALL SORT_MERGE
              ! STATUS = SOR$SORT_MERGE(CONTEXT);
              ! IF NOT .STATUS THEN RETURN .STATUS;
          END;

      ! Put out the statistics, if requested.
      IF .STATISTICS
      THEN
          BEGIN
              STATUS = PRINT_STATS();
              IF NOT .STATUS THEN RETURN .STATUS;
          END;

      ! Call END_SORT to clean up after ourselves
      STATUS = SOR$END_SORT(CONTEXT);
      IF NOT .STATUS THEN RETURN .STATUS;

      ! Return the worst error we've seen
      RETURN .SOR_STS;
      END;
```

								: 0294
	52	0000	0004	00000	.ENTRY	SORENTRY, Save R2		
	5E		CF	9E 00002	MOVAB	CONTEXT, R2		
			08	C2 00007	SUBL2	#8, SP		
		40	A2	D4 0000A	CLRL	SOR_SEV		
44	A2		01	D0 0000D	MOVL	#1, SOR_STS		
	6D	FF3A	CF	9E 00011	MOVAB	COND HAND, (FP)		
			62	D4 00016	CLRL	CONTEXT		
0000V	CF		00	FB 00018	CALLS	#0, INIT STATS		
	3D		50	E9 0001D	BLBC	STATUS, 38		
			5C	DD 00020	PUSHL	AP		
			04	AE 9F 00022	PUSHAB	STATISTICS		
			0C	AE 9F 00025	PUSHAB	SORT_FLAG		
				52 DD 00028	PUSHL	R2		
0000G	CF		04	FB 0002A	CALLS	#4, SOR\$COMMAND		
	2B		50	E9 0002F	BLBC	STATUS, 38		
	0C		04	AE E9 00032	BLBC	SORT_FLAG, 18		
			52	DD 00036	PUSHL	R2		
00000000G	00		01	FB 00038	CALLS	#1, SOR\$SORT_MERGE		
	1B		50	E9 0003F	BLBC	STATUS, 38		
	08		6E	E9 00042	18:	BLBC	STATISTICS, 28	
0000V	CF		00	FB 00045	CALLS	#0, PRINT_STATS		
	10		50	E9 0004A	BLBC	STATUS, 38		
			52	DD 0004D	28:	PUSHL	R2	
00000000G	00		01	FB 0004F	CALLS	#1, SOR\$END_SORT		
	04		50	E9 00056	BLBC	STATUS, 38		
	50		44	A2 D0 00059	MOVL	SOR_STS, R0		
			04	0005D	38:	RET		

: Routine Size: 94 bytes, Routine Base: SCODES + 00B1

```

414      0408 1 ROUTINE INIT_STATS =
415      0409 1
416      0410 1 !++
417      0411 1
418      0412 1 FUNCTIONAL DESCRIPTION:
419      0413 1
420      0414 1 This routine initializes sort/merge statistics.
421      0415 1
422      0416 1 FORMAL PARAMETERS:
423      0417 1
424      0418 1     NONE
425      0419 1
426      0420 1 IMPLICIT INPUTS:
427      0421 1
428      0422 1     NONE
429      0423 1
430      0424 1 IMPLICIT OUTPUTS:
431      0425 1
432      0426 1     NONE
433      0427 1
434      0428 1 ROUTINE VALUE:
435      0429 1
436      0430 1     System status value
437      0431 1
438      0432 1 SIDE EFFECTS:
439      0433 1
440      0434 1     NONE
441      0435 1
442      0436 1 -- BEGIN
443      0437 2 LOCAL
444      0438 2
445      0439 2     STATUS;
446      0440 2
447      0441 2     ! Get the statistics
448      0442 2
449      0443 2     STATUS = SGETJPI(ITMLST=ITMLST);
450      0444 2     IF NOT .STATUS THEN RETURN SOR_ERROR(SORS_SHR_SYSERROR, 0, .STATUS);
451      0445 2     STATUS = SGETTIM(TIMADR=STATS[STAT START]);
452      0446 2     IF NOT .STATUS THEN RETURN SOR_ERROR(SORS_SHR_SYSERROR, 0, .STATUS);
453      0447 2
454      0448 2     STATS[STAT_BUFIO] = .BUFIO;
455      0449 2     STATS[STAT_CPUTIM] = .CPUTIM;
456      0450 2     STATS[STAT_DIRIO] = .DIRIO;
457      0451 2     STATS[STAT_PAGEFLTS] = .PAGEFLTS;
458      0452 2     STATS[STAT_FREPOVA] = .FREPOVA;
459      0453 2
460      0454 2     RETURN SSS_NORMAL;
461      0455 1     END;

```

.EXTRN SYSSGETJPI, SYSSGETTIM

		000C 00000 INIT_STATS:		
53	0000'	CF 9E 00002	WORD	Save R2, R3
		7E 7C 00007	MOVAB	STATS+12, R3
			CLRQ	-(SP)

: 0408  
: 0443

				00000000G	00	00000000*	7E	D4 00009	CLRL -(SP)				
					52		CF	9F 0000B	PUSHAB ITMLST				
					0F		7E	7C 0000F	CLRQ -(SP)				
							7E	D4 00011	CLRL -(SP)				
							07	FB 00013	CALLS #7, SYSSGETJPI				
							50	DD 0001A	MOVL R0, STATUS			0444	
							52	E9 0001D	BLBC STÁTUS, 1\$			0445	
							53	DD 00020	PUSHL R3				
							01	FB 00022	CALLS #1, SYSSGETTIM				
							50	DD 00029	MOVL R0, STATUS				
							52	E8 0002C	BLBS STÁTUS, 2\$			0446	
							52	DD 0002F	PUSHL STATUS				
							7E	D4 00031	CLRL -(SP)				
							03	DD 00033	PUSHL #<<SOR\$ FACILITY@16>+4532>				
							03	FB 00039	CALLS #3, SOR_ERROR				
							04	0003E	RET				
F5 A3		20	FF51	CF	02	14	A3	FO 0003F	INSV BUFI0, #2, #32, STATS+1			0448	
F9 A3		20			02	18	A3	FO 00046	INSV CPUTIM, #2, #32, STATS+5			0449	
					08	20	A3	7D 0004D	MOVQ DIRIO, STATS+20			0450	
					10	28	A3	DD 00052	MOVL FREPOVA, STATS+28			0452	
							50	01	MOVL #1, R0			0454	
								04	0005A	RET			0455

: Routine Size: 91 bytes, Routine Base: SCODES + 010F

```

463 0456 1 ROUTINE ONE_STAT
464 0457 1 (
465 0458 1     CODE
466 0459 1     ): JSB_ONE_STAT =
467 0460 1     ++
468 0461 1     FUNCTIONAL DESCRIPTION:
469 0462 1     This routine gets one sort/merge statistic.
470 0463 1
471 0464 1     FORMAL PARAMETERS:
472 0465 1     CODE.rl.v      Code of statistic to get
473 0466 1
474 0467 1     IMPLICIT INPUTS:
475 0468 1     NONE
476 0469 1
477 0470 1     IMPLICIT OUTPUTS:
478 0471 1     NONE
479 0472 1
480 0473 1     ROUTINE VALUE:
481 0474 1     Value of the statistic
482 0475 1
483 0476 1     SIDE EFFECTS:
484 0477 1     NONE
485 0478 1
486 0479 1     --
487 0480 2     BEGIN
488 0481 2     LOCAL
489 0482 2     RESULT,
490 0483 2     STATUS;
491 0484 2
492 0485 2
493 0486 2     RESULT = 0;
494 0487 2     STATUS = SOR$STAT(CODE, RESULT, CONTEXT);
495 0488 2
496 0489 2     IF NOT .STATUS THEN SOR_ERROR(.STATUS);
497 0490 2
498 0491 2     RETURN .RESULT;
499 0492 2
500 0493 2     END;
501 0494 2
502 0495 2
503 0496 2
504 0497 2
505 0498 1

```

		51 DD 00000 ONE_STAT:		
		PUSHL	R1	0456
		CLRL	RESULT	0492
	0000000G	00	CONTEXT	0493
	00	000004	PUSHAB	
	07	000008	PUSHAB	RESULT
		00000B	PUSHAB	CODE
		00000F	CALLS	#3, SOR\$STAT
		00015	BLBS	STATUS, 18
		00018	PUSHL	STATUS

SORENTRY  
V04-000

M 7  
16-Sep-1984 00:23:12  
14-Sep-1984 13:10:43 VAX-11 Bliss-32 V4.0-742  
[SORT32.SRC]SORENTRY.B32;1

Page 16  
(8)

FF15 CF  
50  
5E 01 FB 0001A  
8E D0 0001F 18:  
04 C0 00022  
05 00025 CALLS #1 SOR-ERROR  
MOVL RESULT, R0  
ADDL2 #4, SP  
RSB

: 0497  
: 0498

; Routine Size: 38 bytes, Routine Base: SCODES + 016A

SOR  
V04

: R

```
507 0499 1 GLOBAL ROUTINE PRINT_STATS =
508 0500 1
509 0501 1 ++
510 0502 1
511 0503 1 FUNCTIONAL DESCRIPTION:
512 0504 1
513 0505 1 This routine prints sort/merge statistics.
514 0506 1
515 0507 1 FORMAL PARAMETERS:
516 0508 1
517 0509 1 NONE
518 0510 1
519 0511 1 IMPLICIT INPUTS:
520 0512 1
521 0513 1 NONE
522 0514 1
523 0515 1 IMPLICIT OUTPUTS:
524 0516 1
525 0517 1 NONE
526 0518 1
527 0519 1 ROUTINE VALUE:
528 0520 1
529 0521 1 System status value
530 0522 1
531 0523 1 SIDE EFFECTS:
532 0524 1
533 0525 1 NONE
534 0526 1
535 0527 1 --
536 0528 2 BEGIN
537 L 0529 2 XIF NOT XDECLARED(COM_K_BPERPAGE)
538 0530 2 XTHEN LITERAL COM_K_BPERPAGE = 512; XFI
539 0531 2
540 0532 2 BUILTIN
541 0533 2 EMUL;
542 0534 2 LOCAL
543 0535 2 FINIS: VECTOR[2];
544 0536 2 CTRSTR: VECTOR[2];
545 0537 2 STATUS;
546 M 0538 2 MACRO
547 M 0539 2 S_(X) =
548 0540 2 (EXTERNAL LITERAL XNAME('SORSK_',X): UNSIGNED(5);
549 0541 2 ONE_STAT(XNAME('SORSK_',X))) X;
550 0542 2
551 0543 2 ! Get the statistics
552 0544 2 STATUS = $GETJPI(ITMLST=ITMLST);
553 0545 2 IF NOT .STATUS THEN RETURN .STATUS;
554 0546 2 STATUS = SGETTIM(TIMADR=FINIS[0]);
555 0547 2 IF NOT .STATUS THEN RETURN .STATUS;
556 0548 2
557 0549 2
558 0550 2 ! Do a quadword subtract to compute the elapsed time.
559 0551 2
560 0552 2 BEGIN
561 0553 2 BIND
562 0554 2 T = STATUS[STAT_START]: VECTOR[2];
563 0555 2 IF .FINIS[0] LSSU :T[0] THEN FINIS[1] = .FINIS[1] - 1;
```

```

564      0556 3     FINIS[0] = .FINIS[0] = .T[0];
565      0557 2     FINIS[1] = .FINIS[1] = .T[1];
566      0558 22    END;
567      0559 22
568      0560 22
569      0561 22    ! Compute the elapsed CPU time, and convert it from 10-millisecond units to
570      0562 22    100-nanosecond units (the standard VMS date/time format) by multiplying
571      0563 22    by 100000.
572      0564 22
573      0565 22    CPUTIM[0] = [CPUTIM[0] - STAT$STAT_CPUTIM];
574      0566 22    EMUL(CPUTIM[0], XREF(100000), XREF(0), CPUTIM[0]);
575      0567 22
576      0568 22
577      0569 22    ! Format and output the statistics
578      0570 22
579      0571 22    CTRSTR[0] = %CHARCOUNT(STR_STATS);
580      0572 22    CTRSTR[1] = UPLIT BYTE(STR_STATS);
581      0573 22    STATUS = SOR$OUTPUT(CTRSTR,
582      0574 22    S_(IDENT),                                ! Address of ASCII ident string
583      0575 22    S_(REC_INP),                            Records input
584      0576 22    S_(LRL_INP),                            Record length
585      0577 22    S_(REC_SOR),                            Records sorted
586      0578 22    S_(LRL_INT),                            Internal record length
587      0579 22    S_(REC_OUT),                            Records output
588      0580 22    S_(LRL_OUT),                            Output record length
589      0581 22    .@SEXTENT,                               Working-set
590      0582 22    S_(NODES),                               Nodes in tree
591      0583 22    (.FREPOVA - .STAT$STAT_FREPOVA)/COM_K_BPERPAGE, ! Memory used
592      0584 22    S_(INI_RUNS),                            Number of runs
593      0585 22    .DIRIO - .STAT$STAT_DIRIO,                Direct I/O
594      0586 22    S_(MRG_ORDER),                           Merge order
595      0587 22    .BUFI0 - .STAT$STAT_BUFI0,                Buffered I/O
596      0588 22    S_(MRG_PASSES),                          Merge passes
597      0589 22    .PAGEF[TS - .STAT$STAT_PAGEFLTS], ! Page faults
598      0590 22    S_(WRK_ALQ),                             Work file allocation
599      0591 22    FINIS[0],                                Wall time
600      0592 22    CPUTIM[0],                                CPU time
601      0593 22    0);                                    Dummy
602      0594 22
603      0595 2     IF NOT .STATUS THEN RETURN SOR_ERROR(SOR$SHR_SYSERROR, 0, .STATUS);
604      0596 2     RETURN $SS_NORMAL;
605      0597 1     END;

```

## .PSECT SPLIT\$,NOWRT,NOEXE,2

74 72 6F 53 20 58 41 56 20 2A 38 31 21 2F 21 00054	P.AAC: .ASCII \!/:!18* VAX Sort/Merge !+Statistics!/:Re\
73 69 74 61 74 53 2B 21 20 65 67 72 65 40 2F 00063	.ASCII \cords read:!12UL!10* Input record length\
55 32 31 21 3A 64 61 65 72 20 73 64 72 6F 63 0007C	.ASCII \!:!9UL!/Records sorted:!10UL!10* Internal\
63 65 72 20 74 75 6E 49 20 2A 30 31 21 4C 0008B	.ASCII \ length:!13UL!/Records output:!10UL!10* \
20 73 64 72 6F 63 65 52 2F 21 4C 55 39 21 3A 000A4	
30 31 21 4C 55 30 31 21 3A 64 65 74 72 6F 73 000B3	
2F 21 4C 55 33 31 21 3A 68 74 67 6E 65 6C 20 000C2	

6-Sep-1984 00:23:12 VAX-11 Bliss-32 V4.0-742  
4-Sep-1984 13:10:43 [SOR132.SRC]SORENTRY.B32;1

Page 19  
(9)

3A	74	75	70	74	75	6F	20	73	64	72	6F	63	65	52	0000DB	
6C	20	64	72	2F	21	20	28	21	24	25	74	30	31	21	0000EA	
72	6F	57	2F			65	74	3A	73	68	74	67	75	4F	0000F4	
20	2A	30	31	21	73	4C	36	21	21	20	6E	65	68	103	0000103	
3A	65	7A	69			65	65	2F	62	20	74	67	67	78	112	0000112
30	31	21	3A	65	62	72	74	21	21	30	55	72	72	53	11C	000011C
6F	20	72	65	62	72	60	60	65	62	30	6C	61	61	31	12B	000012B
65	72	69	44	2F	55	21	21	36	36	69	3A	69	69	55	13A	000013A
30	31	21	4C			69	6F	61	60	2A	21	6E	6E	20	144	0000144
55	39	21	3A	72	64	20	21	55	55	20	30	31	69	20	153	0000153
4F	2F	49	20	64	65	20	21	21	21	30	32	2F	20	162	0000162	
67	72	65	6D	20	56	65	72	75	72	69	70	72	74	16C	000016C	
2F	21	4C	55	36	21	65	65	31	31	66	55	62	61	20	17B	000017B
72	6F	57	20	2A	30	2A	30	31	31	21	4C	65	65	18A	000018A	
69	74	61	63	6F	45	6C	6C	6C	6C	20	55	55	55	194	0000194	
31	21	20	3A	65	70	60	60	69	69	20	20	20	20	20	A3	00001A3
20	64	65	73	70	31	61	61	21	21	20	2A	2A	2A	2A	B2	00001B2
															BC	00001BC
															CB	00001CB
															DA	00001DA
															E4	00001E4
															F3	00001F3
															0202	0000202
															20C	000020C
															21B	000021B
															22A	000022A
															34	0000234

.ASCII \Output record length:!8UL!/Working set e\  
.ASCII \xtent:!6UL!10\* Sort tree size:!14UL!/Vir\  
.ASCII \tual memory:!10UL!10\* Number of initial \  
.ASCII \rns:!6UL!/Direct I/O:!14UL!10\* Maximum \  
.ASCII \merge order:!9UL!/Buffered I/O:!12UL!10\*\\  
.ASCII \ Number of merge passes:!6UL!/Page fault\  
.ASCII \s:!13UL!10\* Work file allocation:!8UL!/E\  
.ASCII \lapsed time: !14%T!7\* Elapsed CPU:!6\* !1\  
.ASCII \4%T\

T=

STATS+12  
SORSK\_IDENT, SORSK\_REC\_INP  
SORSK\_LRL\_INP, SORSK\_REC\_SOR  
SORSK\_LRL\_INT, SORSK\_REC\_OUT  
SORSK\_LRL\_OUT, SORSK\_NODES  
SORSK\_INI\_RUNS, SORSK\_MRGR\_ORDER  
SORSK\_MRGR\_PASSÉS  
SORSK\_WRK\_ALO

.PSECT SCODE\$, NOWRT, 2

			001C	00000
54	D5	AF	9E	00002
53	0000	CF	9E	00006
5E		10	C2	0000B
		7E	7C	0000E
		7E	D4	00010
	0000	CF	9F	00012
		7E	7C	00016
		7E	D4	00018
00000000G	00	07	FB	0001A
	52	50	D0	00021
	10	52	E9	00024
		AE	9F	00027
00000000G	00	01	FB	0002A
	52	50	D0	00031
	04	52	E8	00034
	50	52	D0	00037
			04	0003A

```
ENTRY PRINT STATS, Save R2,R3,R4
MOVAB ONE STAT, R4
MOVAB CPUTIM, R3
SUBL2 #16, SP
CLRQ -(SP)
CLRL -(SP)
PUSHAB ITMLST
CLRQ -(SP)
CLRL -(SP)
CALLS #7, SYSS$GETJPI
MOVL R0, STATUS
BLBC STATUS, 18
PUSHAB FINIS
CALLS #1, SYSS$GETTIM
MOVL R0, STATUS
BLBS STATUS, 28
MOVL STATUS, R0
RET
```

		E8	A3	08	AE	D1	0003B	28:	CMPL	FINIS, T	: 0555
		08	AE	0C	AE	D7	00040		BGEQU	3\$	
		OC	AE	E8	A3	C2	00042	38:	DECL	FINIS+4	
50	E1	A3	20	EC	A3	C2	00045		SUBL2	T, FINIS	0556
		63	63	02	EE	0004F		SUBL2	T+4, FINIS+4	0557	
63		00 000186A0	8F	50	C2	00055		EXTV	#2, #32, STATS+5, R0	0565	
		6E	01E3	63	7A	00058		SUBL2	R0, CPUIM		
		04	AE	0000	8F	3C	00061		EMUL	CPUTIM, #100000, #0, CPUTIM	0566
				CF	9E	00066		MOVZWL	#483, CTRSTR		
				7E	D4	0006C		MOVAB	P.AAC, CTRSTR+4		
					53	DD	0006E		CLRL	-(SP)	
					10	AE	00070		PUSHL	R3	
					51	00G	9A	00073	PUSHAB	FINIS	0592
						64	16	00076	MOVZBL	S^SORSK WRK_ALQ, R1	0591
						50	DD	00078	JSB	ONE_STAT	0590
		7E	OC	A3	F4	A3	C3	0007A	PUSHL	RO	
				51		00G	9A	00080	SUBL3	STATS+24, PAGEFLTS, -(SP)	0589
						64	16	00083	MOVZBL	S^SORSK MRG_PASSES, R1	0588
50	DD	A3	20			50	DD	00085	JSB	ONE_STAT	
	7E		FC	A3		02	EE	00087	PUSHL	RO	
				51		50	C3	0008D	EXTV	#2, #32, STATS+1, R0	0587
						00G	9A	00092	SUBL3	RO, BUFIO, -(SP)	
						64	16	00095	MOVZBL	S^SORSK MRG_ORDER, R1	0586
		7E	08	A3	F0	50	DD	00097	JSB	ONE_STAT	
				51		A3	C3	00099	PUSHL	RO	
						00G	9A	0009F	SUBL3	STATS+20, DIRIO, -(SP)	0585
						64	16	000A2	MOVZBL	S^SORSKINI_RUNS, R1	0584
50	7E	10	A3	F8	A3	C3	000A6	JSB	ONE_STAT		
				50 00000200	8F	C7	000AC	PUSHL	RO		
				51		00G	9A	000B4	SUBL3	STATS+28, FREPOVA, R0	0583
						64	16	000B7	DIVL3	#512, RO, -(SP)	
						50	DD	000B9	MOVZBL	S^SORSK NODES, R1	0582
						00G	9A	000BE	JSB	ONE_STAT	
					51	64	16	000C1	PUSHL	RO	
						50	DD	000C3	MOVZBL	S^SORSK LRL_OUT, R1	0581
					51	00G	9A	000C5	JSB	ONE_STAT	0580
						64	16	000C8	PUSHL	RO	
					51	50	DD	000CA	MOVZBL	S^SORSK REC_OUT, R1	0579
						00G	9A	000CC	JSB	ONE_STAT	
					51	64	16	000CF	PUSHL	RO	
						50	DD	000D1	MOVZBL	S^SORSK LRL_INT, R1	0578
					51	00G	9A	000D3	JSB	ONE_STAT	
						64	16	000D6	PUSHL	RO	
					51	50	DD	000D8	MOVZBL	S^SORSK REC_SOR, R1	0577
						00G	9A	000DA	JSB	ONE_STAT	
					51	64	16	000DD	PUSHL	RO	
						50	DD	000DF	MOVZBL	S^SORSK LRL_INP, R1	0576
					51	00G	9A	000E1	JSB	ONE_STAT	
						64	16	000E4	PUSHL	RO	
					51	50	DD	000E6	MOVZBL	S^SORSK REC_INP, R1	0575
						00G	9A	000E8	JSB	ONE_STAT	
					51	64	16	000EB	PUSHL	RO	
						50	DD	000ED	MOVZBL	S^SORSK IDENT, R1	0574
						AE	9F	000EF	JSB	ONE_STAT	
									PUSHAB	RO	
										CTRSTR	0573

0000G	CF	15	FB 000F2	CALLS #21, SOR\$OUTPUT	
	52	50	DD 000F7	MOVL R0, STATUS	
	10	52	E8 000FA	BLBS STATUS, 4\$	
		52	DD 000FD	PUSHL STATUS	
		7E	D4 000FF	CLRL -(SP)	
FF34	C4	00000000*	8F DD 00101	PUSHL #<<SORT\$ FACILITY@16>+4532>	
		03	FB 00107	CALLS #3, SOR_ERROR	
			04 0010C	RET	
		50	01 DD 0010D	4\$: MOVL #1, R0	
			04 00110	RET	0596

: Routine Size: 273 bytes, Routine Base: \$CODE\$ + 0190

606	0598	1		
607	0599	1	END	
608	0600	0	ELUDOM	

## PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	72	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
SPLITS	567	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	673	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

## Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	27	0	581	00:01.0
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	20	3	252	00:00.6

## COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:SORENTRY/OBJ=OBJ\$:\$ORENTRY MSRC\$:\$ORENTRY/UPDATE=(ENH\$:\$ORENTRY)

: Size: 673 code + 639 data bytes  
 : Run Time: 00:15.6  
 : Elapsed Time: 00:50.7  
 : Lines/CPU Min: 2312  
 : Lexemes/CPU-Min: 22959

SORSENTRY  
V04-000

: Memory Used: 127 pages  
: Compilation Complete

F 8  
16-Sep-1984 00:23:12 VAX-11 Bliss-32 v4.0-742

Page 22

SO  
VO

0364 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

